

FINAL PROJECT

CALCULATING CHURN RATES

CODECADEMY

LEARN SQL FROM SCRATCH

Alejandro Martinez | Dec 10th, 2018



CHALLENGE

Codeflix, a streaming video startup, is interested in measuring their user churn rate four months after launch.

The marketing team is particularly interested in comparing churn rates between two user segments that represent users who were acquired through two distinct channels.

Thanks to my knowledge of SQL, I've been tasked to provide management with the data they need.

To get to the final answer I needed to understand how the table looks, and how much information it can provide.

These are the questions that once answered, provided the data I needed to complete the challenge.

1. Get Familiar with the Company

- A. How many months has the company been operating?
- B. Which months do I have enough information to calculate a churn rate?
- C. What segments of users exist?

2. What is the overall churn trend since the company started?

3. Compare churn rates between user segments.

- D. Which segment of users should the company focus on expanding?

A. How many months has the company been operating?

1. First I needed to visualize the table. This query showed me there are 4 columns; id, subscription_start, subscription_end, and segment.

A: Because the first user registered on December 1st 2016 and the last user registered on March 30th 2017, we can conclude the company has been operating for 5 months (including April).

```
SELECT *
FROM subscriptions
LIMIT 100;
```

Query Results			
id	subscription_start	subscription_end	segment
1	2016-12-01	2017-02-01	87
2	2016-12-01	2017-01-24	87
3	2016-12-01	2017-03-07	87
4	2016-12-01	2017-02-12	87

B. Which months do I have enough information to calculate a churn rate?

2. Then, I searched for the subscription start date of the first user by using MIN, and the subscription start date of the last user registered by using MAX.

B: We can only calculate the churn rate for 3 months; January, February and March. December is not considered because it was the first month and users can't cancel before 30 days; neither April because we don't have data for new users that signed up in April.

```
SELECT MIN(subscription_start)
FROM subscriptions;

SELECT MAX(subscription_start)
FROM subscriptions;
```

Query Results	
min(subscription_start)	
2016-12-01	
max(subscription_start)	
2017-03-30	

C. What segments of users exist?

- 1. To answer this question I created a simple query that showed the column segment results grouped by each segment.

```
SELECT segment
FROM subscriptions
GROUP BY segment;
```

segment
30
87

C: There are 2 customer segments, 30 and 87.

2. What is the overall churn trend since the company started?

CHURN = CANCELLATIONS / TOTAL SUBSCRIBERS

```
WITH months AS
(
  SELECT
    '2017-01-01' AS first_day,
    '2017-01-31' AS last_day
  UNION
  SELECT
    '2017-02-01' AS first_day,
    '2017-02-28' AS last_day
  UNION
  SELECT
    '2017-03-01' AS first_day,
    '2017-03-31' AS last_day
),
```

```
cross_join AS
(
  SELECT *
  FROM subscriptions
  CROSS JOIN months
),
```

```
status AS
(
  SELECT id, first_day as month,
    CASE
      WHEN (subscription_start < first_day)
      AND (subscription_end > first_day
      OR subscription_end is NULL)
      THEN 1
      ELSE 0
    END as is_active,
    CASE
      WHEN (subscription_end BETWEEN first_day AND last_day)
      THEN 1
      ELSE 0
    END as is_canceled
  FROM cross_join),
```

```
status_aggregate AS
(
  SELECT month,
    SUM(is_active) AS sum_active,
    SUM(is_canceled) AS sum_canceled,
  FROM status
  GROUP BY month
)
```

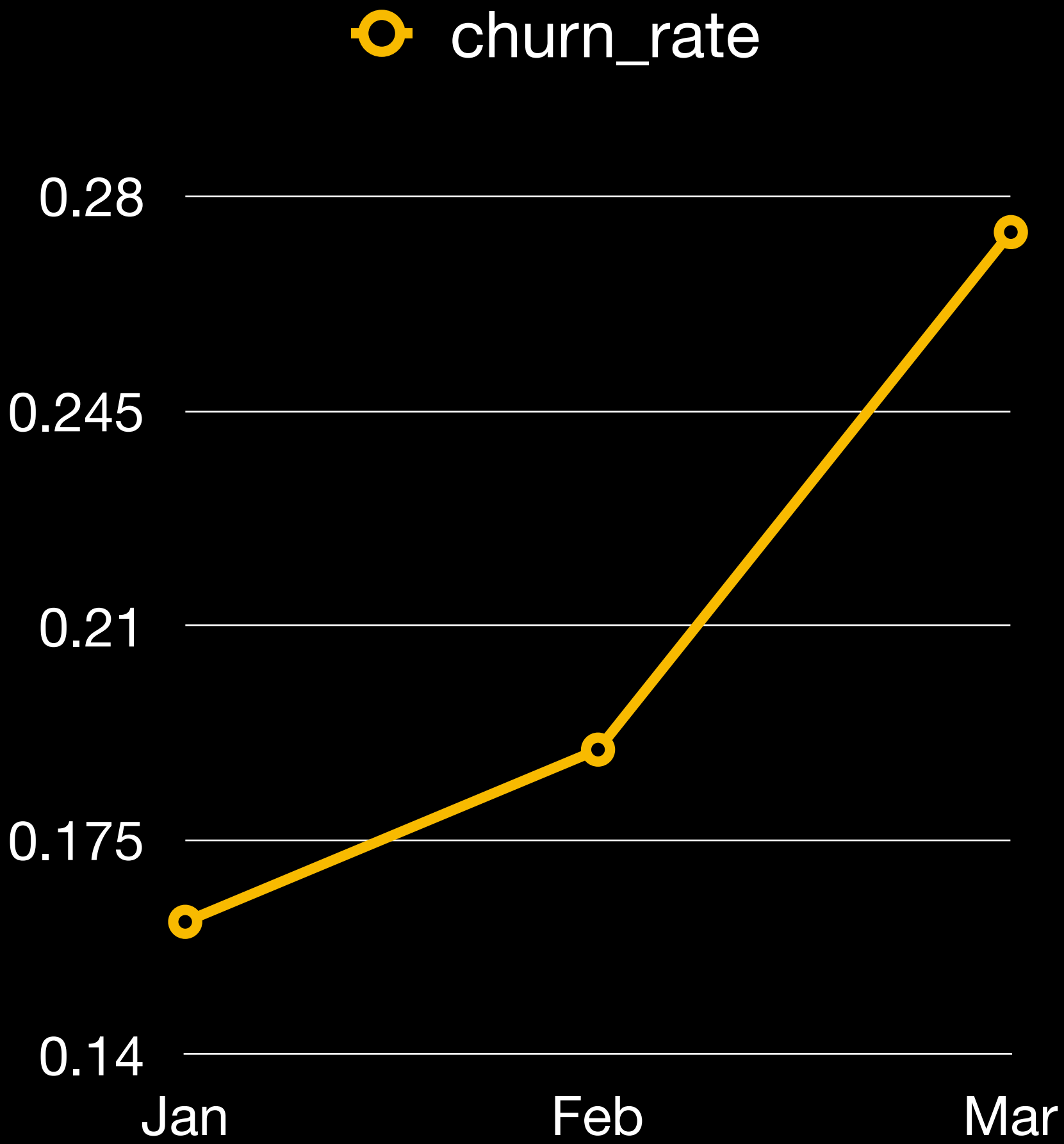
```
SELECT month,
  1.0 * sum_canceled/sum_active AS churn_rate
FROM status_aggregate;
```

1. To answer this question I created temporal the tables months, cross_join, status and status_aggregate.
2. These tables helped me organize the data so I could calculate the churn rate in a much easier way.

2. What is the overall churn trend since the company started?

CHURN = CANCELLATIONS / TOTAL SUBSCRIBERS

month	churn_rate
2017-01-01	0.161687170474517
2017-02-01	0.189795918367347
2017-03-01	0.274258219727346



2: The total churn rate has been increasing since January

3. Compare churn rates between user segments.

1. To compare churn rates between both segments I edited the previous query (total churn rate) and added the following CASE scenarios
 1. When segment 87 is active
 2. When segment 87 is cancelled
 3. When segment 30 is active
 4. When segment 30 is cancelled
2. Finally I calculated churn rates for each segment.

```
status AS
(
  SELECT id, first_day as month,
    CASE
      WHEN (subscription_start < first_day)
      AND (subscription_end > first_day)
      OR subscription_end is NULL
      AND segment = 87
      THEN 1
      ELSE 0
    END is_active_87,
    CASE
      WHEN (subscription_end BETWEEN first_day AND last_day)
      AND segment = 87
      THEN 1
      ELSE 0
    END is_canceled_87,
```

Segment 30 is not shown here to fit the code in the slide.

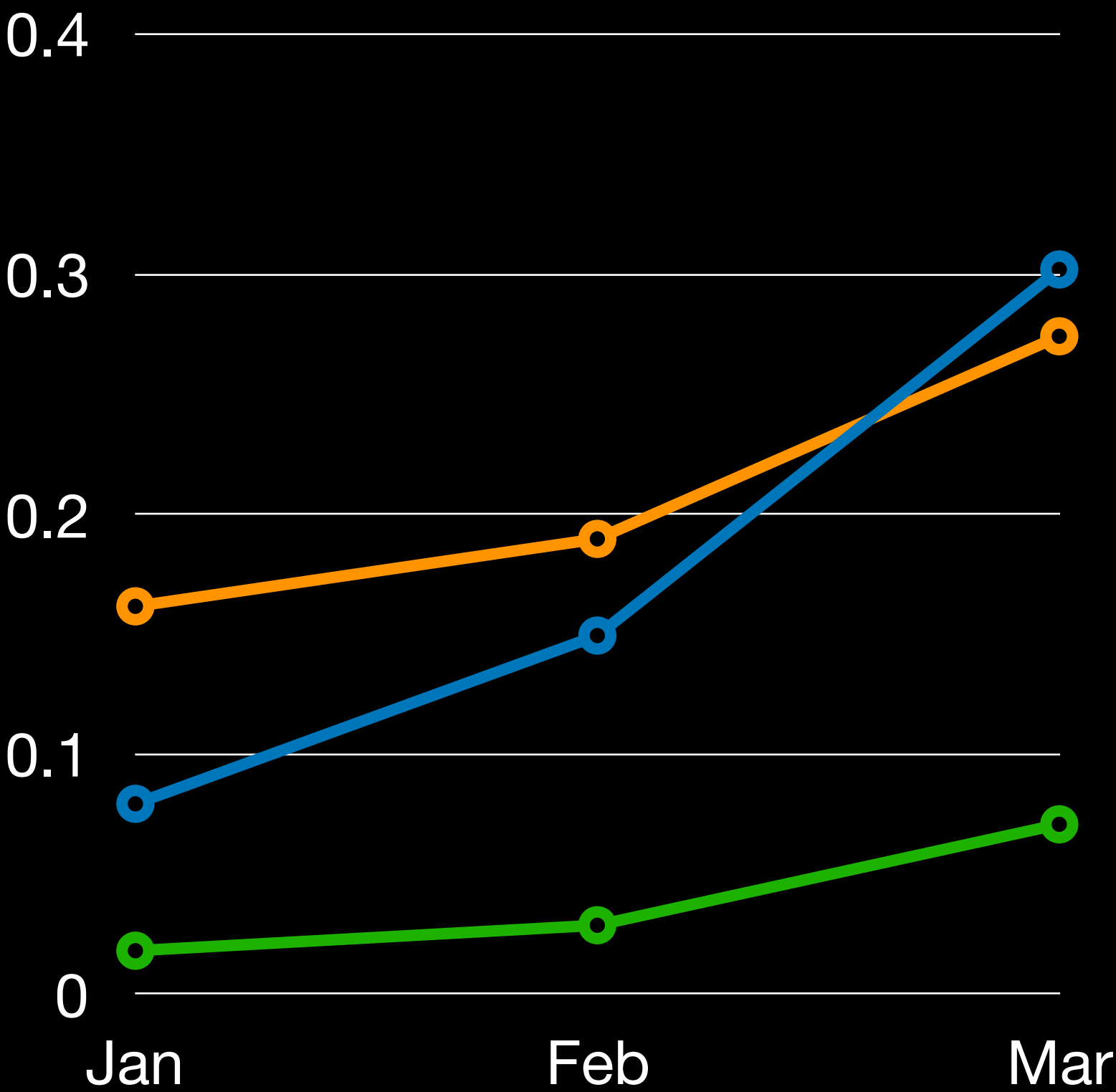
```
status_aggregate AS
(
  SELECT month,
    SUM(is_active_87) AS sum_active_87,
    SUM(is_active_30) AS sum_active_30,
    SUM(is_canceled_87) AS sum_canceled_87,
    SUM(is_canceled_30) AS sum_canceled_30
  FROM status
  GROUP BY month
)
```

```
SELECT month,
  1.0 * sum_canceled_87/sum_active_87 AS churn_rate_87,
  1.0 * sum_canceled_30/sum_active_30 AS churn_rate_30
FROM status_aggregate;
```


3. Compare churn rates between user segments.

Here I compare each segment's churn rate vs the total churn rate.

month	churn_rate	churn_rate_87	churn_rate_30
2017-01-01	0.161687170474517	0.251798561151079	0.0756013745704467
2017-02-01	0.189795918367347	0.32034632034632	0.0733590733590734
2017-03-01	0.274258219727346	0.485875706214689	0.11731843575419



FINAL CONCLUSION

The company should focus in expanding segment 30, as it that segment shows a smaller churn rate.

- churn_rate_87
- churn_rate_30
- tot_churn_rate